UNIVERSIDAD NACIONAL EXPERIMENTAL DELTACHIRA
VICERECTORADO ACADÉMICO
DEPARTAMENTO DE INGENIERÍA EN INFORMÁTICA

PROGRAMA DE SISTEMAS DE CONTROL

				Especialidad: Ingeniería en Informática			
				Densidad Horaria			Pre-requisito
Código	Nombre de la Asignatura	Estrategias Metodológicas	Unidades Crédito	H.T.	H.P.	T. hrs/sem.	8206T 8207T
8208T	Sistemas De Control	Ver anexo	06	4	2	6	

INTRODUCCIÓN

El modelo abstracto matemático de computación es de gran importancia debido a que el conocimiento de la terminología, notación, y técnicas de la teoría de computación enfatizando la computabilidad se utilizan para el estudio de inteligencia artificial, diseño de computadores, análisis de algoritmos. El estudiante adquiere una gran cantidad de herramientas y habilidades, siendo las más importantes: la manipulación de gramáticas de contexto libre, procesos recursivos como parte de un procedimiento, calculo lambda, Decidibilidad e indecidibilidad y demostraciones matemáticas de cada proceso, finalmente sé efectúa un análisis de complejidad. El enfoque de esta materia está dirigido hacia el aprendizaje de Autómatas Finitas, Lenguajes Formales y Maquinas de Turing específicamente el estudio de las posibilidades o capacidades que pueden tener los computadores hoy o en el futuro, lo cual exige que se adquieran conocimientos de teoría de conjuntos, lógica, manipulaciones abstractas matemáticas, circuitos digitales. Este contenido fue dictado en los semestres 2000-11, 2001-1, 2001-II y 2002-1, siendo luego reemplazado por el contenido de Sistemas de Control.

OBJETIVO GENERAL

Formar alumnos que sean capaces de aplicar los conocimientos adquiridos en el área de Autómata Finita, lenguajes Formales y Maquinas de Turing para desarrollar aplicaciones útiles en el área del hardware y del software, que sean capaces de aplicar los mecanismos formales a problemas prácticos, fundamentalmente a la definición léxica, sintáctica y semántica de los lenguajes de programación, construir programas analizadores léxicos a partir de los mecanismos formales estudiados lo que les permite sentar las bases formales para abordar la implementación de compiladores e interpretes a futuro.

OBJETIVOS ESPECÍFICOS

- Lenguajes regulares a través del estudio de Autómatas Finitas.

- Se analizan los lenguajes de contexto libre ó independientes de contexto lo cual introducen las nociones necesarias para la posible implementación de compiladores.

- Se introducen lenguajes recursivos enumerables y el diseño de máquinas de Turing y de Post para analizar los diferentes conceptos de computabilidad.

DESCRIPCIÓN SINÓPTICA DE LA ASIGNATURA

Fundamentos de análisis matemático. Alfabetos y lenguajes. Definiciones Recursivas. Lenguajes Regulares, Teorema de Kleene, no-determinismo, Lenguajes No regulares, Decidibilidad, Gramáticas de Contexto libre. Gramáticas Regulares, Forma Normal de Chomsky, Autómatas de Pila, Lenguajes no libre de contexto, intersección y complemento, Parsing , Decidibilidad, Maquinas de Turing, Maquinas de Post, Variantes de las Maquinas de Turing, Lenguajes Recursivos Enumerables. Codificación de las Maquinas de Turing, Jerarquía de Chomsky, Computabilidad, Complejidad, Computadores.

METODOLOGÍA

La metodología recomendada para dictar la materia es la siguiente:

- Clases teóricas
- Dinámicas de Grupo
- Resolución de ejercicios
- Trabajos de Investigación
- Desarrollo de proyectos de envergadura considerable.

Los instrumentos a ser utilizados por el docente son:

- Pizarrón
- Proyector de transparencias
- Data Show o proyector de multimedia
- Computadora personal
- Láminas de Power Point
- Internet

UNIDAD I. FUNDAMENTOS MATEMÁTICOS			
OBJETIVO: Al finalizar la unidad, el alumno, interpretara los conceptos fundamentales de Lógica, Teoría de Conjuntos, diferentes métodos de demostraciones Matemáticas			
OBJETIVOS ESPECÍFICOS	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACION**
Al finalizar la unidad el alumno estará en condiciones de: > Distinguir los elementos de la Lógica Elemental > Conocer las definiciones básicas. > Efectuar operaciones con conjuntos. > Manipular diferentes tipos de demostraciones, Inducción, Contradicción.	> Lógica Elemental: Tablas de Verdad, equivalencia, negación, conjunción, reciproca, contrapuesta, frase Abierta, cuantificación universal, existencial. > Definiciones básicas. > Definición de Conjunto Potencia. > Operaciones con Conjuntos. > Relaciones Y Funciones. > Principio de Inducción Matemática. > Contradicción. > Cardinalidad. > Manejo de Apuntadores	Exposición Didáctica. Participación espontánea y sugerida. Torbellino de Ideas. Asistir a las secciones de teoría y laboratorio. Discutir los temas tratados, los problemas y soluciones propuestas. Búsqueda de información en Internet. Recurrir a la bibliografía recomendada para afianzar los Conocimientos impartidos.	

UNIDAD II. EXPRESIONES REGULARES, TEOREMA DE KLEENE			
OBJETIVO: Al finalizar la unidad, el alumno construirá definiciones recursivas de Expresiones regulares, Demostrará la equivalencia entre expresiones regulares, grafos de transición y Autómata Finitas.			
OBJETIVOS ESPECÍFICOS	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACION**
Al finalizar la unidad el alumno estará en condiciones de: > Distinguir las diferentes formas de manipular Lenguajes. > Construir definiciones recursivas de Expresiones Regulares. >Construir Autómatas Finitas. > Analizar / Demostrar la equivalencia entre expresiones regulares, autómatas finitos, gratos de transición.	>Alfabetos. >Lenguajes. >Operaciones con Cadenas. >Operaciones con Lenguajes > Definiciones Recursivas > Expresiones Regulares > Autómatas Finitas. > Gratos de Transición. > Teorema de Klecne. Equivalencia entre Autómatas Finitas y Gratos de transición. Equivalencia entre gratos de transición y Expresiones Regulares. Equivalencia entre Expresiones Regulares Y Autómatas Finitas	Exposición Didáctica. Participación espontánea y sugerida. Asistir a las secciones de teoría y laboratorio. Elaborar los proyectos propuestos en clase. Investigar en Internet. Recurrir a la bibliografía recomendada para afianzar los conocimientos impartidos	Pruebas Cortas

UNIDAD III. LENGUAJES REGULARES			
OBJETIVO: Al final la unidad, el alumno analizará, definirá Lenguajes Regulares, construirá autómatas finitas deterministas y no deterministas; podrá demostrar, decidir si un Lenguaje dado es o no regular			
OBJETIVOS ESPECÍFICOS	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACION**
Al finalizar la unidad el alumno estará en condiciones de: > Convertir autómatas finitas deterministas en no-deterministas y viceversa, minimizar autómatas. > Diseñar circuitos utilizando autómatas finitos con salida(maquinas de estado) > Aplicar el lema de Bombeo para demostrar si un lenguaje dado pertenece al conjunto de lenguajes regulares o no. > Decidir la equivalencia entre Lenguajes.	>Autómatas Finitas Deterministas. >Autómatas Finitas no-deterministas. >Autómatas Finitas con salida. > Maquina de Mealy. > Maquina de Moore > Equivalencia entre maquinas de Mealy y Maquinas de Moore. > Conversión de maquina de Mealy a maquinas de Moore y viceversa. > Lenguajes Regulares. > Lenguajes no-regulares. > Lema de Bombeo. > Decibilidad.	Exposición Didáctica. Discusión dirigida. Participación espontánea y sugerida. Torbellino de Ideas. Asistir a las secciones de teoría y laboratorio. Investigar en Internet Construir proyectos pertinentes tales como analizadores léxicos, brazos reboticas etc. Recurrir a la bibliografía Recomendada para afianzar los conocimientos impartidos.	

UNIDAD IV. AUTÓMATA DE PILAS, GRAMÁTICAS DE CONTEXTO LIBRE			
OBJETIVO: Al finalizar la unidad, el alumno analizara, definirá y construirá Gramáticas regulares, de contexto libre, autómatas de pila, aplicara la forma normal de Chomsky, demostrara la equivalencia entre CFG y PDA			
OBJETIVOS ESPECÍFICOS	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACION**
Al finalizar la unidad el alumno estará en condiciones de: > Definir gramáticas de contexto libre, Gramáticas regulares. > Distinguir si una gramática es ambigua o No. > Convertir a la forma normal de Chomsky. > Analizar la equivalencia entre autómatas de pila y las gramáticas de Contexto libre.	>Gramáticas de contexto libre (CFG.) >Gramáticas regulares. >Derivación por derecha y por izquierda. > Árboles. > Ambigüedad. >Forma Normal de Chomsky (CNF.) >Forma normal de Backus (BNF.) > Autómata de pilas (PDA). > Equivalencia entre las gramáticas de contexto libre y los autómatas de pila. Sea CFG =PDA.	Exposición Didáctica. Discusión dirigida. Participación espontánea y sugerida Discutir los temas tratados, los problemas y soluciones propuestas. Elaborar los programas propuestos para verificar ambigüedad, convertir las diferentes formas nórmales.	Entrega de Proyectos

UNIDAD V. AUTÓMATAS DE PILAS, LENGUAJES DE CONTEX TO LIBRE			
OBJETIVO: Al finalizar la unidad, el alumno analizara, definirá y construirá lenguajes de contexto libre, lenguajes no de contexto libre, intersección, complemento, Parsing, pertinencia a CFL, Decidibilidad.			
OBJETIVOS ESPECÍFICOS	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACION**
Al finalizar la unidad el alumno estará en condiciones de: > Definir los lenguajes de contexto libre. > Distinguir si un lenguaje dado pertenece o no al conjunto de lenguajes de contexto libre. > Analizar, interceptar lenguajes y Encontrar el complemento, la Derivación de cualquier Gramática dada. > Decidir si dos CFG diferentes definen el mismo lenguaje, si existe ambigüedad, si el complemento de una CFG dada es de contexto libre.	> Lenguajes de Contexto libre CFL. > Lenguajes no de contexto libre. > Lema de Bombeo para lenguajes de Contexto libre. > Intersección. > Complemento. > Parsing. > Decidibilidad. > Algoritmo de John Cocke, Tadao Kasami y Daniel H. Younger. Conocido como CYK	Exposición Didáctica. Discusión dirigida. Participación espontánea y sugerida. Asistir a las secciones de teoría y laboratorio. Elaborar los programas para interceptar, hallar el complemento de cualquier CFL dado. Diseñar un Parser. Investigar en Internet sobre compiladores, interpretes. Recurrir a la bibliografía recomendada para afianzar los conocimientos impartidos.	Entrega de Proyectos Parcial I (30%)

UNIDAD VI. TEORÍA DE MAQUINAS DE TURING			
OBJETIVO: Al finalizar la unidad, el alumno analizara, definirá, construirá, codificara maquinas de Turing maquinas de Post, teorema de Minsky, variaciones de las maquinas de Turing, lenguajes recursivos enumerables, Jerarquía de Chomsky, Complejidad, Computadoras.			
OBJETIVOS ESPECÍFICOS	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACION**
Al finalizar la unidad el alumno estará en condiciones de: > Definir, Codificar maquinas de Turing, maquinas de Posl. > Analizar las variaciones de las maquinas de las maquinas de Turing, Lenguajes recursivos enumerables. > Codificar maquinas de Turing, distinguir Jerarquía de Chomsky, definir analizar Computadoras..	> Maquinas de Turing. > Maquinas de Post. > Teorema de Minsky. > Variaciones / variantes de las maquinas de Turing. > Lenguajes Recursivos Enumerables. > Codificación de las maquinas de Turing. > Jerarquía de Chomsky. > Complejidad. > Computadoras.	Exposición Didáctica. Discusión dirigida. Participación espontánea y sugerida. Diseñar programas que simulen Maquinas de Turing, maquinas de Post. Investigar en Internet. Recurrir a la bibliografía Recomendada para afianzar los conocimientos impartidos.	Entrega de Proyectos Parcial II (35%) Conformado por la entrega de proyectos. Parcial III (35%)

BIBLIOGRAFÍA.

Deán Kelley: Teoría de Autómatas y Lenguajes Formales, Prentice Hall 1995

J. Glenn Brookshear: Teoría de la Computación, lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana 1993

T. A. Sudkamp: Introduction to the theory ofComputer Science, Addison-Wesley Publishing Company 1997

Linz P.: Introduction to Formal Languajes and Autómata, D. C. Heath and Company 1990

Ullman J. D.; Hopkroft, J. E.: Introduction to Autómata Theory, Languajes and Computations. Addison-Wesley Publishing Company 1979

Dijkstra, E. W., editor.. Formal Development Of Programs And Proofs,Mó'ison-\Nes\e^ Publishing Company, 1990

Manna, Z.; Waldinger, R., The Logical Bases For Computing Programming, Addison-Wesley Publishing Company 1993.

Kfoury, A.J.; Molí, R. N.; Arbib, M. A., A Programming Approach to Computability, Springer-Verlag, 1982

Daniel L. A. Cohén: Introduction to Computer Theory, John Wiley & Sons Inc. 1986

Henry Ledgard, Michael Marcotty: The Programming Languaje Landscape, Science Research Associates Inc. 1981